

A METHOD AND SYSTEM FOR MANAGING A DIRECTORY WITH A TEMPLATE

RELATED APPLICATION

5 This application claims priority to U.S. provisional application number 60/310,529, titled, "Configuring the Heirarchical Structure of Directory Entries/Managing Composite Directory Object", filed August 6, 2001, the contents of which are herein incorporated by reference.

10 FIELD OF THE INVENTION

The present invention is in the field of directory management, specifically aimed at managing the directory with a template.

BACKGROUND OF THE INVENTION

15 Computer networks often contain numerous resources, varying in type, size, and use. These resources include computers and peripheral devices. In order to consistently identify such resources and facilitate their use by users, networks provide directory services. Directories normally contain information about the resources, (i.e., email addresses on the network). Ideally, a directory service will hide the physical implementation of the network it is provided on, representing the various resources of the network in a consistent manner. 20 This has the advantage of allowing users to address any resource on a network without knowledge of the actual physical configuration of the network, the resource, or its connection to the network.

25 One of the most important directory services in use today is the Lightweight Directory Access Protocol (LDAP). The LDAP directory service model is based on entries. Each entry in LDAP comprises a collection of attributes. In turn, each attribute consists of a type and one or more values. The type of an attribute identifies the format of the data contained in the values of that attribute. Thus, an attribute with the type "mail", for example, might contain one or more email addresses in its value fields. In addition to attributes, each entry also has a 30 unique name, called a "distinguished name". The distinguished name makes possible unambiguous references to the entry.

Entries in an LDAP directory service are organized in a tree structure. Under the protocol, data may be stored in any arbitrary fashion, as long as a tree structure is maintained. Presently, there are no well-defined protocols for structuring Directory Information Trees (DITs). This may not be a problem for an application that sets up its own directory and populates it from scratch, since the application can define the structure of its DIT in any manner. When an application has to work with an existing directory, however, it will have to first become familiar with the DIT structure in use. And this may prove to be a daunting task, especially in the case of large and complex directories.

Aside from the issue of familiarizing new applications with existing directory structures, there is also the problem of imposing standard structures throughout a directory. Presently, if an organization wishes to impose such a standard, it must do so through an administrator who would need to manually maintain the structure of the DIT each time an entry is added, deleted, or modified.

Accordingly, there exists a need for a method and system for managing a directory so that when entries are created in the directory, the rest of the defining structure can be built automatically.

BRIEF SUMMARY OF THE INVENTION

The present invention may include a method for managing a directory comprising the steps of creating a template to define the directory comprising one or more definitions corresponding to one or more object types, wherein for a given one of said object types, said corresponding definition identifies zero or more others of said object types that should be automatically created and added to the directory whenever said given object type is added to the directory; creating an object of said given type; adding said object of said given type to the directory; automatically creating zero or more objects of said others of said object types; and adding said zero or more objects to the directory. In one aspect of the invention, the zero or more objects of others of the object types are added as children to the object of the given type in the directory. In another aspect, the one or more object types are Java object types.

In another aspect of the invention, the one or more definitions comprise one or more attributes, and in another aspect, the one or more attributes have values. In another aspect,

the step of automatically creating the zero or more objects comprises the steps of selecting one of the definitions that correspond to the given type of object that is created; and reading the zero or more others of the object types from the selected definition. In another aspect, the template is created using a markup language, and in another aspect the markup language is an extensible markup language and/or a generalized markup language.

The present invention may also include a template for managing a directory comprising one or more definitions corresponding to one or more object types, wherein for a given one of said object types, said corresponding definition identifies zero or more others of said object types of zero or more objects that should be created whenever an object of said given object type is added to the directory.

In one aspect of the invention, the one or more object types are Java object types. In another aspect, the one or more definitions comprise one or more attributes. And in yet another aspect, the one or more attributes have values.

The present invention may also include a method for managing a directory of one or more entries comprising the steps of creating a template to provide one or more instruction on how to manage the directory; creating at least one of the entries; and adding said at least one entry to the directory in accordance with at least one of said instructions.

In one aspect, the preferred embodiment further comprises the step of deleting said at least one entry from the directory in accordance with one of the instructions. In another aspect, the one or more instructions comprise one or more definitions corresponding to one or more entry types wherein for a given one of said entry types, the corresponding definition identifies zero or more of said entries that must be added to the directory whenever an entry of said given type is added to the directory.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a sub-tree of a typical directory.

FIG. 2 shows a sub-tree of a directory with the templates of the present invention in place.